WL-TR-95-1153

FEATURE AND EXTRACTOR EVALUATION CONCEPTS FOR AUTOMATIC TARGET RECOGNITION (ATR)

TIMOTHY D. ROSS
LORI A. WESTERKAMP
DAVID A. GADD
ROBERT B. KOTZ

TARGET RECOGNITION BRANCH
SENSOR ATR TECHNOLOGY DIVISION
2241 AVIONICS CL
BUILDING 620, RM C3-J75
WRIGHT-PATTERSON AFB, OH 45433-7318

OCTOBER 1995

FINAL REPORT FOR PERIOD 01 JANUARY 1995 – 01 OCTOBER 1995

SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE US GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS REPORT IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Timothy D. Ross
Project Engineer
Target Recognition Branch

Dale E. Nelson, Chief
Target Recognition Branch
Sensor ATR Technology Division

CLYDE R. HEDDINGS, Major, USAF
Deputy, Sensor ATR Technology Division
Sensors Directorate

Do not return copies of this report unless contractual obligations or notice on a specific document require its return.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	October 1995	Final, 01/01/1995 – 10/01/1995

4. TITLE AND SUBTITLE

FEATURE AND EXTRACTOR EVALUATION CONCEPTS FOR AUTOMATIC TARGET RECOGNITION (ATR)

5. FUNDING NUMBERS

C: N/A
PE: 63226E
PR: ARPA
TA: MS
WU: 01

6. AUTHOR(S)

TIMOTHY D. ROSS
LORI A. WESTERKAMP
DAVID A. GADD
ROBERT B. KOTZ

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

TARGET RECOGNITION BRANCH
SENSOR ATR TECHNOLOGY DIVISION
2241 AVIONICS CL
BUILDING 620, RM C3-J75
WRIGHT-PATTERSON AFB, OH 45433-7318

SVERDRUP
4200 COLONEL GLENN HWY
SUITE 500
BEAVERCREEK, OH 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

SENSORS DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7318
POC: Timothy D. Ross, AFRL/SNAT, (937) 255-1115 x4045

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

WL-TR-95-1153

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
Approved for public release; distribution unlimited.	

13. ABSTRACT *(Maximum 200 Words)*

This report develops concepts that will support the evaluation planning for the MSTAR features and feature extractors. These concepts will be used later in building a detailed evaluation plan. We began our development by distinguishing between the evaluation of a feature set and the evaluation of an extractor. The specifics for feature evaluation depend upon whether or not it is meaningful to define a truth-value; but in either case, features are evaluated in terms of their sensitivity (at first individually and then as a set) to various "factors". The factors of interest fall into the categories of Known, Class, and Noise. Ideal features would be discriminating (high sensitivity to class factors), robust (low sensitivity to noise factors), and predictable (predictable sensitivity to known factors). The evaluation of extractors (including auxiliary information such as runtime/memory use estimates and feature uncertainty) is based on accuracy (when meaningful), design quality, and good software engineering principles.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Automatic Target Recognition, ATR, Features, Feature Extractor, Performance Assessment, Evaluation Methodology	32
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Contents

List of Figures

List of Tables

Acknowledgements

The authors would like to acknowledge the technical contributions made to this report by Ron Dilsavor, Sverdrup, regarding mocked-up truth-based features, Lou Tamburino, WL/AAAT, on the Algorithm Development versus Performance Estimation Product Team's roles, and Mark Minardi, WL/AARA, on the truth-based versus extractor-based features discussion. We also acknowledge the management guidance provided by Capt. Tom Burns, WL/AARA, and Ed Zelnio, WL/AARA, and the secretarial support of Becky Haupt, WL/AARA, and Becky Dixon, SelectTech.

Chapter 1

Introduction

The Moving and Stationary Target Acquisition and Recognition (MSTAR) program will develop a modular, model-driven target recognition system for high resolution Synthetic Aperture Radar (SAR) surveillance imagery. The program's goals include robust recognition of large numbers of targets, articulated targets, occluded targets and variously configured targets. Additionally, the program will focus on the development of a distributed, collaborative ATR development environment based on Khoros free-access image processing environment.

The MSTAR Program is funded by the Advanced Research Projects Agency (ARPA) and is managed by Wright Laboratory (WL/AARA). The MSTAR architecture is defined in the MSTAR program documentation. Briefly though, the MSTAR system is partitioned into six modules: Focus of attention, Indexing, Search, Predict, Extract and Match (see Figure 1.1).

This report proposes a general evaluation methodology for the features and their extractors within the MSTAR program. An overview of the extract module is given in Chapter 3. This report provides technical background and does not represent an official position on the MSTAR evaluation plan or procedures.

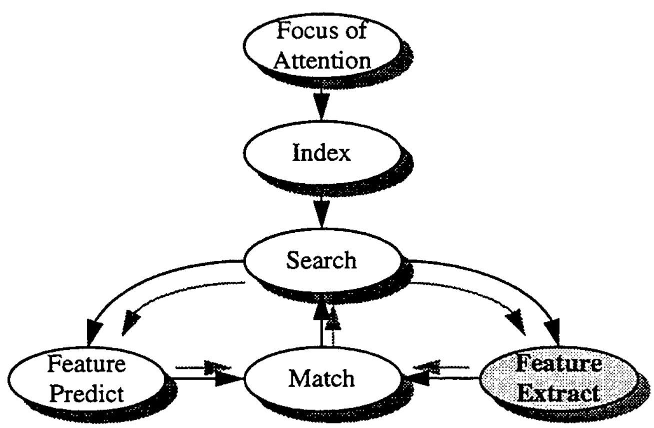

Figure 1.1: MSTAR Architecture

This page intentionally blank.

Chapter 2

Concepts and Roles

This chapter will discuss key concepts and a preliminary position on the various roles for those involved in the development and evaluation of the features and their extractors. Sorting out these concepts and their relationships is an important first step in defining an evaluation methodology.

2.1 Concepts

First of all, we would like to differentiate the concept of "feature" from its "extractor," extractors from their "module," modules from their "contractors," and, finally, contractors from the Feature Extraction "Team."

Features are, for example, sets of peaks, texture, and target dimensions. There is a fuller discussion of features below, but we need to introduce features here as a key primitive concept. In terms of evaluation, we are looking for features that are discriminating and robust.

Extractors are the algorithms or software that compute a feature from an image. There may be one or more extractors per feature, depending on the type of feature and options for its extraction. In terms of evaluation, we want extractors to be accurate and efficient.

A Feature Extraction *Module* consists of a collection of extractors, an executive, an estimator for computer resource usage, and possibly one or more functions that compute distances between features. Modules are evaluated in terms of their design and software engineering properties, and the accuracy of the computer resource usage estimates.

The Feature Extraction Module *Contractors* are ERIM and SAIC-Tucson. Each contractor will develop a module, but those modules may have many common extractors and common features. Each contractor will develop extractors that may be used in both contractor's modules. Modular extractors are a general goal, but must be pursued only to the extent that they do not increase development costs or artificially constrain the feature design. Each contractor will participate in the Feature Extraction Team that will lead in the development and evaluation of the MSTAR feature

set. Contractors will be evaluated in terms of their ability to cooperate within the Feature Extraction Team and the MSTAR team as a whole. The contractor's demonstrated ability to define and evaluate new features is clearly important, as is their ability to develop and evaluate extractors. Note the distinction being made between the evaluation of a contractor and the evaluation of the other concepts.

MSTAR will develop the feature set as a team, perhaps with the Systems Integrator taking the lead. However, we expect the Feature Extraction *Team* (including ERIM, SAIC-Tucson, and Wright Lab) to take the lead in recommending new features and providing the sensitivity analysis to support feature selection.

2.1.1 Features

We have had considerable debate about what exactly *defines* a feature. In one approach, features are defined by their extraction method; a feature is, by this definition, what a feature extractor extracts. This puts the entire burden on Predict to produce the same feature; that is, any difference between extracted and predicted features is an error in the prediction. Extractors do not make "errors" in the features that they produce. Of course, extractors do not get a free ride. Instead of accuracy, extractors are evaluated directly in terms of the ability of the features to contribute to classifier performance.

An alternative approach, based on "truth" features, would be to distinguish the evaluation of the *feature* from that of the feature *extractor*. In this approach, features are defined by ground truth or human interpretation for measured images or directly from the model for synthetic images. Extractors are then *estimators* of that true state.

Both approaches have advantages and disadvantages. The extraction-based approach is nice and clean-cut. It does not require the introduction of the somewhat artificial truth state. It can be applied to all the known features.

On the other hand, the truth-based approach better supports the MSTAR program goal of ATR module commodities and leads to a more tractable evaluation methodology. However, it is not always possible to define a meaningful truth state for every feature (e.g. peaks and texture).

Confidence in the extracted features can have the usual meaning when the features are estimates (i.e. truth-based features). For the extractor-based features, an interpretation of confidence must be developed between Extract, Predict, and Match. This is being addressed by the MSTAR Uncertainty Tiger Team which was formed in August 1995.

Some features may be defined in the truth-based sense, but only within the context of an assumed model for the data from which the feature is extracted. For example, superresolution approaches to feature extraction may assume a damped complex exponential model for SAR data. For these approaches, feature extraction is equivalent to model parameter estimation. In reality, measured (and often predicted) SAR data does not exactly match the assumed model and hence the "true" feature is not

well-defined nor available for feature extraction performance analysis. In order to use a truth-based approach in this case it is necessary to synthesize data according to the assumed model, apply the feature extractor, and analyze performance by comparing extracted features with the true features used to synthesize the data. While we get some of the benefits of using the truth-based approach for these features, it introduces a new problem. To what degree does the assumed model fit actual SAR imagery? The effect of model mismatch is an important new problem when using the truth-based approach to performance analysis for model-based parameter estimation features.

How these concepts for a "feature" translate into evaluation methodologies is discussed in Section 4.1.1. Our recommendation, at the moment, is to use the truth-based approach whenever possible and the extract-based approach otherwise.

2.1.2 Factors

A concept that we have struggled to label is the collection of all the things that go into making an image what it is.[1] For example, the image depends on the type of target present, its position, orientation and articulations. The image depends on the image collection and formation parameters. The image also depends on a myriad of effects that we generally characterize as noise, including speckle and calibration errors. Here we call the entire collection of things that determine an image the "factors."

We partition these factors into three sets. One set we call Class Factors (CF). This includes all the factors that are unknown and we are attempting to determine. Example members of CF include target type and orientation. We will need to distinguish the Hypothesized CF (HCF) and the True CF (TCF).

The second set we call Known Factors (KF). This includes all factors that influence the image but are known during the ATR process. Example members of KF include squint and depression angles.

The third set we call Noise Factors (NF). This includes all factors that are unknown and we do not attempt to determine. Example members of NF include actual detector noise and target details such as dents or minor articulations.

These three sets of factors are involved in the PEM loop as shown in Figure 2.1. It will be useful in later discussions to refer to the combination of CF and NF as simply the Unknown Factors (UF). Note that some extracted factors may be treated as UF initially and then as KF later in the process.

2.2 Roles

Within Wright Lab's support structure for MSTAR, there are two Product Teams (PT) involved in the feature set selection, Algorithm Development (AD) and Performance Evaluation (PE). The analysis of features for selection, which is an AD

[1]This concept is important in Section 4.1.1.

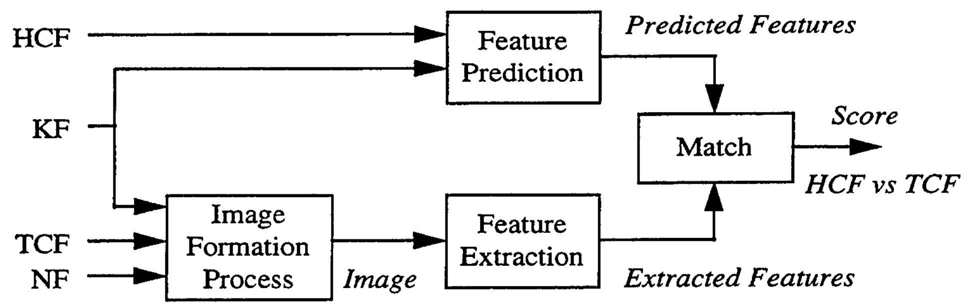

Figure 2.1: Factor's Role In PEM Loop

function, is not entirely separate from the evaluation of the feature set, which is a PE function. Although this report is a product of the MSTAR PE PT, our discussion of feature evaluation will overlap considerably with the feature selection analysis to be performed as part of the AD PT. We recognize that much of the analysis we consider "feature evaluation" may be done under the AD PT and that PE PT's role in feature evaluation may be limited to spot checks, filling any gaps, resolving conflicting results, and those tests that especially need independence.

To summarize, consider the following activities:

- new feature invention — this is an AD PT function that will be performed by all MSTAR participants, with major contributions from the Feature Extraction contractors.

- extractor implementation — this is an AD PT function performed by the Feature Extraction contractors. We expect cooperation between ERIM, SAIC-Tucson, and the Government in decisions on who implements what extractor, including extractors for features contributed from outside the Feature Extraction (FE) Team.

- feature selection — this involves both the AD and PE PTs. The sensitivity analysis to support selection is primarily the responsibility of the FE team. The System Integration team is expected to lead the decision process that considers the sensitivity analysis and other factors, such as impacts on system wide efficiency. We expect cooperation within the FE Team on how the sensitivity analysis is performed and who analyzes which features. The PE PT will probably not directly evaluate the feature set, instead evaluate the sensitivity analysis that went into selecting the feature set.

- FE Module implementation — this is an AD PT function which can be performed by ERIM and SAIC-Tucson with relative independence.

- Self and Independent Evaluations — these are PE PT functions performed by contractors and the Government.

Chapter 3

Module Overview

Roughly speaking, the Feature Extraction module takes an image chip and instructions from Search and provides a set of features back to Search. We expand on the description of this module by first defining how we use the term "feature" and then considering the inputs, outputs and functionality. These properties are key to developing an evaluation methodology for feature extraction.

3.1 Module Inputs

The module inputs consist of a Region-of-Interest (ROI), a list of desired features, and related parameters.

The ROI consists of an image array (with dimensions), sub-ROI delimitation using masks or polygons, and attributes of the sub-ROIs. These attributes include labels (such as water, forest, urban) and confidences in the labels.

The desired feature list is provided by the Search Module to avoid the computational cost of finding features that are not needed.

The relevant parameters include feature parameters (such as thresholds or windows to use in computing features) and sensing parameters (such as image acquisition and formation parameters).

3.2 Module Outputs

The output is, of course, features. When reporting values and locations for any feature, the confidence in those values and locations will also be reported. MSTAR defines three classes of features: local, regional and global. Local features include brightness peaks, other structures and texture. The regional features are labeled sub-ROIs (e.g. object, shadow, no-show, occluded, and surround). The global features include target orientation and dimensions.

ERIM expands on this baseline set of features by using Image Relational Trees (IRT) to represent local and regional features (see [4]). Other specific features that will

be explored by ERIM and sub-contractor, The Ohio State University (OSU), includes frequency characteristics of scattering centers and multi-resolutional texture/target characteristics. ERIM also plans to monitor the literature for emerging features that could help the MSTAR program; therefore, we should expect the feature set to evolve over the life of the program. Because of the unique character of the IRT, ERIM's Feature Extraction module will compute the distance between two IRTs, as might be requested by Search or Match modules.

SAIC has a large library of existing feature extraction primitives. SAIC plans to develop new features using genetic algorithms. They also plan to do a significant amount of preprocessing (clutter suppression and gain normalization) that will be coordinated with FOA.

In addition to the features themselves, the Search Module may be interested in an estimate of the computer resources that would be required to compute some feature set. The Feature Extraction module is responsible for providing this estimate.

The feature set will change over the course of the program. However, it is difficult to develop the evaluation methodology without some specific features to consider. This has lead us to define a "strawman" feature set. This set will allow us to begin consideration of the evaluation problem, but we do not intend for this set to influence the feature set *design* in any way. The evaluation methodology will be modified and expanded to address the actual feature set that the Design Product Team produces. The strawman feature set consists of:

- Local Features
  - Peak Set
  - Texture
- Regional Features
  - Labeled sub-regions (e.g. object, shadow, no-show)
- Global Features
  - Target Dimensions
  - Target Orientation
  - Target Position

The original gray-scale image is considered a "feature" that Predict may be required to produce; but its extraction is trivial and not an evaluation consideration here.

3.3 Functionality

The Feature Extraction module computes the features (or distance or runtime estimate) as requested by the Search module. The Feature Extraction module is defined by the MSTAR design documentation.

Chapter 4

Performance Measures

We define the performance measures[1] in three categories: Functional Performance, Design, and Software Engineering. Functional Performance is concerned with how well the module performs its basic function. This is the most important category because it is objectively measurable and highly relevant to the MSTAR program goals. The Design is concerned with the fundamental algorithms and their development. Although more difficult to assess, this is also highly relevant to the overall program. Software Engineering is concerned with the software implementation of the algorithms and, although not of direct relevance to the MSTAR program, it is essential to building an understanding of the Functional Performance and Design.

4.1 Functional Performance

Functional performance can be broken into four categories for feature extractor evaluation:

- impact on overall system classification performance,
- impact on overall system efficiency,
- factor sensitivity, and
- accuracy of non-feature products (i.e. extractor run time estimates and feature confidences).

Evaluation of the impact on overall system performance (classification and efficiency) will be considered within the System Integration Evaluation.

This leaves feature sensitivity and the accuracy of non-feature outputs.

[1]We will occasionally use the phrase "output variable" in place of performance measure. This is a reflection of the fact that performance measures are often the *output* of experimental evaluations.

4.1.1 Sensitivity

Features are "good" if they help the classifier. The classifier decides on a class based on the "closeness" of the extracted features to some reference features (in this case, predicted from a model). Features make classification easier if they increase the distance between features from images representing different classes and decrease the distance between features from images representing the same class. Therefore, a key factor in the evaluation of features is the relative change in features for a between or within class change in the image.

The concept of a "factor" was introduced in Section 2.1.2. There were Class Factors (CF), Known Factors (KF), and Noise Factors (NF). The CF could be True (TCF) or Hypothesized (HCF) (see Figure 2.1).

Ideally the feature comparison score would always reflect the difference between HCF and TCF and be independent of NF and KF. So, we want features with high sensitivity to changes in CF, low sensitivity to changes in NF, and whose variation due to KF can be predicted. That is, we want the features to be sensitive to between-class variations (aka discriminating, inter-class sensitive) and insensitive to within-class variations (aka robust, intra-class insensitive, stable).

Note that CF (and therefore NF) will change during the PEM loop's search. That is, initially CF may not include the target's variant or precise pose. So, at this point in the search, it is desirable that features *not* be sensitive to target variant or fine pose. As the search converges and we are down to determining target variant and precise pose, it becomes desirable that the features be sensitive to variant and pose. In addition to target type and pose, CF may include region labels (including object and background). Obstructions, configurations, articulations, and CC&D may also become part of CF, depending on the overall system design.

The method for assessing sensitivity depends on the approach used to define the feature.

Extract-Based Features

Sensitivity is measured by the amount the feature changes for a given change in the factors. Generally the feature sets will be compared with the Match Module. Figure 4.1 summarizes this basic approach.

The particular method for measuring "change in feature" will depend on the type of feature. As examples, a method for each of the types in the strawman feature set will be discussed. The change in peaks could be measured by the Match module's score in comparing the two sets of peaks. The method for measuring the change in texture will depend on how texture is characterized. If texture is characterized by a number then a simple difference could be used. Change in region labels could be measured as "same" or "different." Change in target dimensions could be the numeric difference in each dimension.

Similarly, the methods for measuring changes in factors will depend on the particular factor. As examples, a method for measuring changes in factors will be proposed

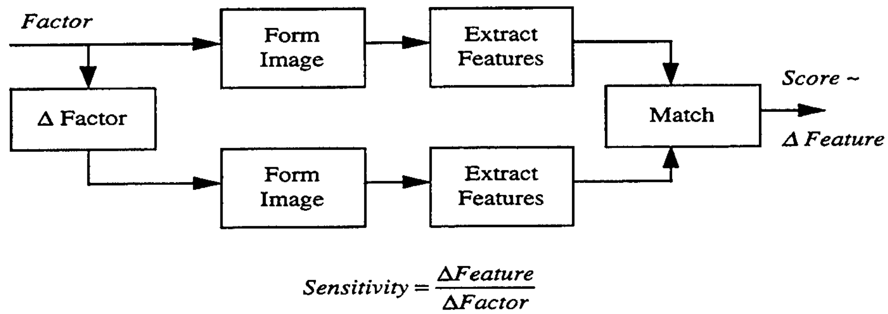

Figure 4.1: Feature Sensitivity Assessment

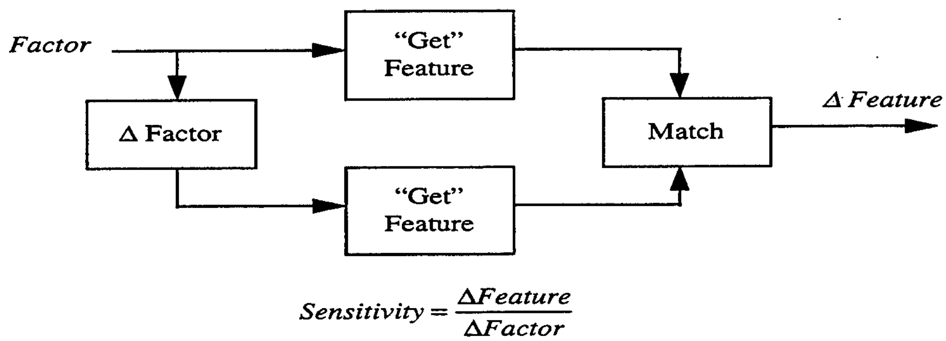

Figure 4.2: "Truth" Feature Sensitivity Assessment

for some of the obvious class components. Target type is the principal component for our purposes. The method for measuring change will be to label two types as "same" or "different," with a similar method for target class. The change in orientation can be measured as a simple numerical difference, similarly for any simple numeric component of class, such as degree of obstruction, squint angle, or depression angle. Articulation can often be characterized by a number.

Truth-Based Features

For some features we can introduce an artificial "truth" feature that can be gotten directly from the CF. For example, target length is available from the class model. Therefore, rather than using the methodology of Figure 4.1 which requires two image formations and two extractions, we can use the method of Figure 4.2 and determine sensitivity by getting the feature directly from the class description. The sensitivity of these "truth" features can be determined relatively easily because it does not require image formation or feature extraction. Also, the sensitivity of a "truth" feature is independent of the particular extraction method (and vendor); therefore, it can be determined just once and that value used over the entire life of the program.

We can then indirectly assess the sensitivity of the actual features by comparing

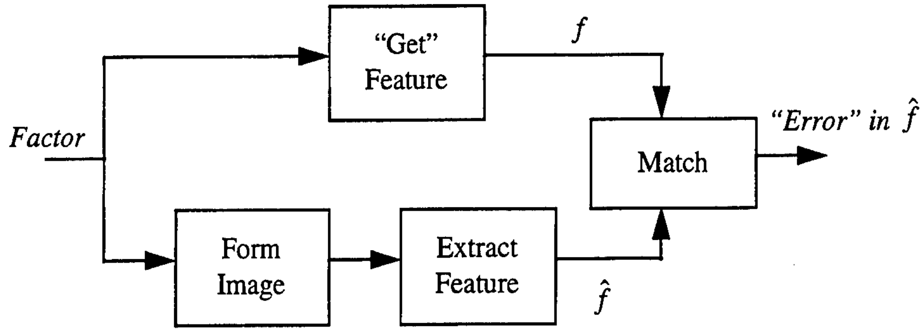

Figure 4.3: Feature "Accuracy" Assessment

them to the "truth" features, as shown in Figure 4.3. This "accuracy" assessment can also be made with relative ease. Therefore, when it is meaningful to do so, we can use this two step process to expedite the evaluation of a feature's sensitivity.

As discussed in the definition of features on page 4, these truth-based sensitivities must be interpreted with care and should only be used for features that are reasonably thought of as estimates of some well defined property. Further, the magnitude of a sensitivity must be interpreted relative to the accuracy with which the feature can be extracted. For example, exact target length may be very discriminating, but length with an uncertainty of ±20 feet would not be.

With respect to the strawman feature set, we expect to be able to use the truth-based assessment method for region labels, target dimensions and target orientation. We expect to use the more expensive extractor-based assessment method for the features based on peaks and texture.

Sensitivity Aggregation Methods

Depending on the results of the sensitivity analyses for individual features, various combinations of features may be assessed. Similarly, the analysis with respect to individual factors may be expanded to consider combinations of factors. The many methods for formally aggregating sensitivity results will be assessed and applied as appropriate. Methods to be considered include those identified in References [2, Chapters 5-9], [3, Chapters 9 and 10], and [5, pp.465-497].

4.1.2 Accuracy Assessment Methods

With the truth-based method, feature evaluation becomes a more conventional evaluation of accuracy. We now consider how the accuracy of various properties will be assessed.

The properties that are involved in features include things such as:

- location,
- value/magnitude/strength/amplitude,

	A	B
a	12	1
b	3	9

Table 4.1: Region Label Confusion Matrix

- extent/area,
- uncertainty,
- other discrete and continuous characteristics, and
- Abstract Data Structures (ADS).

For the strawman feature set defined on page 8, extractor runtime, and feature distances, the attributes of interest include:

- Peak (location, extent, magnitude-value, uncertainty)
- Peaks (ADS)
- Texture (location, extent, value, uncertainty)
- Labeled sub-regions (location, extent, label-value, uncertainty)
- Target Dimensions (length-value, width-value, uncertainty)
- Target Orientation (value, uncertainty)
- Extractor Runtime (value)
- Feature Distance (value, uncertainty)

We now develop more specific measures for labels and confidences.

Labels

For labeled regions, the "truth" data consists of image chips with sub-regions labeled with their true values. The difference between the labels associated with each pixel in a truthed image chip and those from extract will be recorded in a confusion-matrix. The matrix will have dimensions $n \times n$, where n is the number of possible labels. There will be a confusion matrix entry for each pixel in each image chip. Table 4.1 is the confusion matrix that would result for the example image in Figure 4.4.

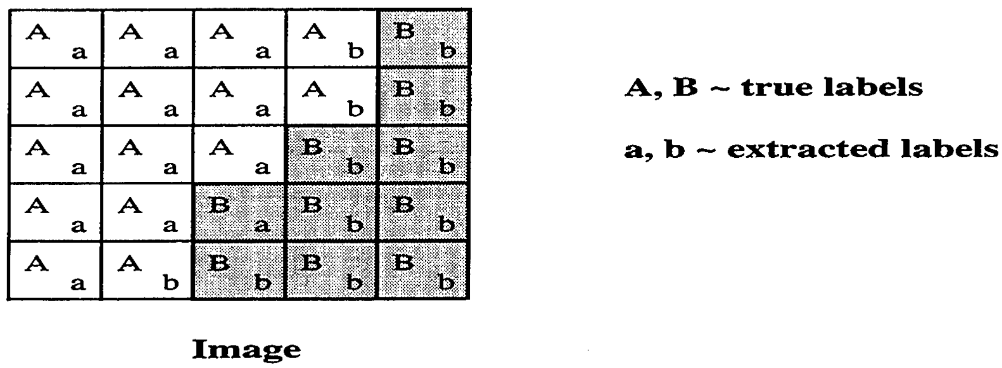

Figure 4.4: Image with True and Extracted Labels

Confidences

The key to evaluating extractor provided confidences is to define "truth" data in a meaningful way. For this discussion we consider just two types of confidence expressions. Other types will be dealt with as they arise.

One type of confidence is related to labels and is expressed as a probability. This confidence probability is meant to reflect the relative frequency with which the declared label is in fact the true label. A second type of confidence expression is related to numerical values. In this case there are two parameters. One parameter (ϵ) gives the size of an interval and the second parameter (δ) gives the probability that the true values lie within that interval, i.e.

$$P(|x - \hat{x}| \leq \epsilon) \geq 1 - \delta,$$

where x is the true value, $\hat{x}$ is the estimate, ϵ is half of the interval size, and $1 - \delta$ is the confidence.

So, in either case, we are getting the probability that some assertion is true (i.e. that the label is correct or that a number falls within an interval). Therefore, a histogram of the data collected, with the bins defined on some interval in probability, will be used. Within each bin of the histogram, the fraction of time that the assertion was true constitutes the "truth" data. That is, this is our best knowledge of the true probability that the assertion was correct. The limits of the bin are the estimated uncertainty and are *accurate* to the extent that they match the fraction of time that the assertion was actually true. Figure 4.5 represents some notional results.

Consider target orientation as a particular example. There may be some fixed confidence, say 99%, and a varying interval. For each chip we get an extracted orientation and a confidence interval. The true orientation is either within that confidence interval or it is not. After extracting the orientation from a large number of image chips, say 1000, we have some relative frequency of success (true orientation was within the confidence limit of the extracted orientation), say 98.7%. The accuracy of the extractor's confidence output is reflected in the difference between these two

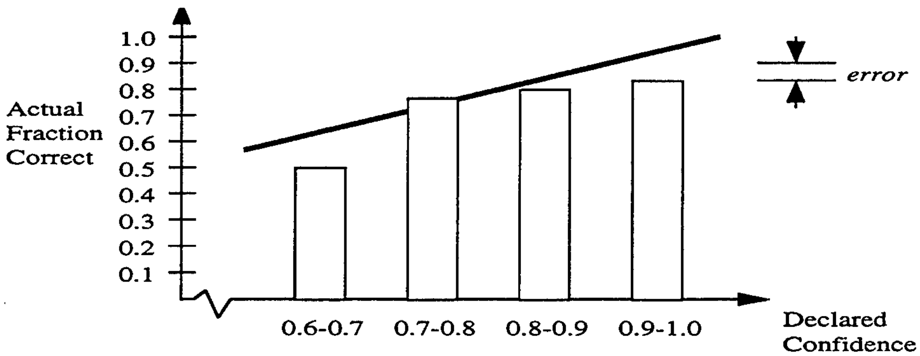

Figure 4.5: Confidence Histogram

percentages. If they are close then the confidence is accurate; if they are not close then the confidence is not accurate. For region labels, the data that went into the confusion matrix discussed above can be used to build similar histograms with similar confidence evaluation implications.

4.1.3 Summary

In summary, with respect to the strawman feature set, we expect to use an extractor-based method to asses the sensitivity of the peak and texture based features. We expect to assess the sensitivity of "truth" features for region labels, target dimensions, and target orientation. We expect to evaluate the "accuracy" of the extracted estimates of these "truth" features. Finally, we expect to assess the accuracy of any confidence measure associated with features and the extractor's estimates of its own runtime. The only output unaccounted for is the feature distance function that an extractor may provide. While no direct evaluation is planned for this function, it will be an integral part of the sensitivity assessments since Match will be used to compare features and, presumably, Match will use the extractor's feature distance function.

4.2 Design

The Design is assessed in terms of proofs/derivations, modifiability of the algorithms (as opposed to software modifiability), and efficiency of the algorithms.

The clarity, correctness and completeness of derivations and proofs used to support the module's design, uncertainty estimation and timing estimates will be assessed. The traceability of assertions to the literature will be an important consideration.

Modifiability of the algorithms is directly related to the MSTAR program goal of "commodization" of ATR algorithms and the evaluation of model-driven ATR system development. We are concerned about modifiability with respect to new and/or modified:

- adjacent modules,

- targets,
- sensors/sensor parameters/image properties,
- scenarios/missions/area of operation, and
- features.

The MSTAR extended operating conditions are especially relevant modifications. Modifiability will be assessed by the amount of "effort" required to make these changes *and* the quality of the results after modification. The "effort" required will be measured in terms of number of changes, person-hours used, and amount of data/computer resources used. The number of parameters and their tuning methods will be considered in this assessment, along with the degree to which the changes are model-driven versus data-driven.

Efficiency of the underlying algorithms is considered here. The degree of concurrancy/parallel operation possible for the algorithm is also of interest.

As for extractor-specific considerations, the feature set and the extractor should require *no* changes for changes in targets or scenarios. The changes with respect to sensor parameters and image properties should be systematic or even automated. The extractor's scalability is of concern with respect to chip size, image properties (e.g. complex or polarimetric), and feature resolution as requested by Search.

4.3 Software Engineering

We use as goals the categories from Reference [1, pp.24-27], i.e. modifiability, efficiency, reliability, and understandability. To these we add portability — an especially important issue for the MSTAR teaming arrangement. The SI contractor will probably have the best insight into this after doing the first integration.

- Modifiability: Modifiability is considered here, with respect to software engineering, and again in Design, with respect to the overall process of modifying the module. Modifiability with respect to the EOCs is particularly relevant.

- Efficiency: Efficiency is also consider here and again above with respect to Design. Here we are concerned with the use of time and memory resources by the delivered software. The impact of each feature on the computation resource requirements of Extract, Predict, and Match is an important consideration.

- Reliability: The ability of the software to handle weird data sets and degrade gracefully is of interest. The correctness of the software (i.e. how faithfully it implements the specified algorithm) will be assessed. Tools such as *purify* and *profile* will be used to look for memory leaks, unnecessary use of disks, etc.

- Understandability: The documentation should be clear, accurate, and up-to-date. The completeness will be assessed, including a mathematical description of the algorithms and any undocumented features. A subjective judgement will be made as to the ease-of-use of the documentation. The code should also be understandable, making good use of modularity and abstraction, and be well commented. Standard metrics, such as average function size, will be considered.

- Portability/Rehost Verification: The degree of difficulty encountered in rehosting the software is of interest, not only to the degree required for us to get things running, but also as a measure of how well we are supporting the "commodization" of ATR modules. This will be reflected in the amount of work required for the rehost and whether or not we were able to get it compiled and running on all of our systems. The rehost will include exercising *all* options available to the user, looking for bombs and the ability to duplicate vendor provided test cases.

This page intentionally blank.

Chapter 5

Summary

This report develops concepts that will support the evaluation planning for the MSTAR features and feature extractors. These concepts will be used later in building a detailed evaluation plan. We began our development by distinguishing between the evaluation of a feature set and the evaluation of an extractor. The specifics for feature evaluation depend upon whether or not it is meaningful to define a truth-value; but is either case, features are evaluated in terms of their sensitivity (at first individually and then as a set) to various "factors." The factors of interest fall into the categories of Known, Class, and Noise. Ideal features would be discriminating (high sensitivity to class factors), robust (low sensitivity to noise factors) and predictable (predictable sensitivity to known factors). The evaluation of the extractors (including auxiliary information such as runtime/memory use estimates and feature uncertainty) is based on accuracy (when meaningful), design quality, and good software engineering principles.

This page intentionally blank.

Bibliography

[1] Grady Booch. *Software Engineering with Ada.* The Benjamin Cummings Publishing Company, Redwood City, California, 1983.

[2] Pierre A. Devijver and Josef Kittler. *Pattern Recognition: A Statistical Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[3] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, New York, second edition, 1990.

[4] John F. Stach and Scott W. Shaw. Robust relational trees by scale-space filtering. In *Proceedings of the IEEE Signal Processing Time-Frequency and Time-Scale Conference*, September 1992. Victoria, B.C., Canada.

[5] Henry Stark and Robert O'Toole. Statistical pattern recognition using optical Fourier transform features. In Henry Stark, editor, *Applications of Optical Fourier Transforms*, chapter 11, pages 465–497. Academic Press, New York, 1982.